The standard data model approach to patient record transfer

Kip Canfield, PhD, Marcelo Silva, BS
Laboratory for Healthcare Informatics
University of Maryland, UMBC

Kerry Petrucci, PhD CRNP
Department of Physiological Nursing
University of Washington, Seattle

ABSTRACT

This paper develops an approach to electronic data exchange of patient records from Ambulatory Encounter Systems (AESs). This approach assumes that the AES is based upon a standard data model. The data modeling standard used here is IDEF1X for Entity/Relationship (E/R) modeling. Each site that uses a relational database implementation of this standard data model (or a subset of it) can exchange very detailed patient data with other such sites using industry standard tools and without excessive programming efforts. This design is detailed below for a demonstration project between the research-oriented geriatric clinic at the Baltimore Veterans Affairs Medical Center (BVAMC) and the Laboratory for Healthcare Informatics (LHI) at the University of Maryland.

INTRODUCTION

Patient records exist to improve patient care, ease administrative and financial reporting, and allow research access to healthcare data. Computer-based patient records (CPR) have advantages in each of these areas. Electronic patient records are more accessible to providers and therefore can improve continuity of care [1]. Henrion [2] estimates that the cost savings from information technology would be 4 percent of the total health care costs by the year 2000. Cross-patient research access to patient records is prohibitively expensive except with database records. An important theoretical advantage and practical problem for CPR is transfer of patient record information over networks. This is needed for creation of research repositories, quality control for patient care and billing, and consultation by remote providers. Evaluation of this CPR transfer demonstration project supports the following propositions concerning the standard data model approach to patient record transfer and replication.

Proposition 1 - Information Density

The information density required in a well designed AES is only possible with some type of data model. This implies that transfer of a patient record should contain the data model information. A data model contains much implicit information about the enterprise. For example, in an E/R model that is meant to be implemented on a relational database management system (RDBMS), all of the relationship data is implicit in the foreign keys of the ultimate relational tables. This information would be difficult to code for transfer of a patient record without reference to the data model. An example of this relationship-oriented information would be a link between each physician order and the patient problem that it addresses. Such a link would allow audits of all resources expended for a specific problem.

Proposition 2 - Efficiency and Reliability

Industry standard tools already exist for certain data modeling techniques and relational database architectures and they offer an efficient way of enabling the implementation of the standard data model patient record transfer. The industry standards for IDEF1X data modeling [3] and RDBMSs have sufficient penetration that a rich set of interoperable, multivendor tools for CASE (Computer-Aided Software Engineering), scripting methods, and querying with a standard language such as Structured Query Language (SQL) are available. Examples of these tools are used in the demonstration project and evaluated.

Proposition 3 - Communications

Existing methods of communications on the Internet, such as e-mail and FTP are sufficient to implement the standard data model patient record transfer approach for complex text data. The use of industry standard database tools produce text script files that are easily compressed, encrypted, and transferred on the Internet. This allows developers to largely automate the routine tasks associated with patient record transfer without regard to vendor or platform.

Each of these propositions is argued to be supported by this patient record transfer demonstration project. The data model used documents an AES that is used for real patient data and is in regular use at the Baltimore site. The patient record transfers are real patient data that have been made anonymous for the purposes of this project. Each procedure described in the project methodology is tested in a realistic simulation and discussed below.

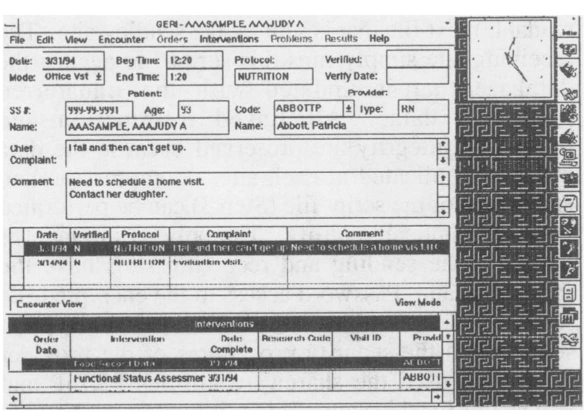
AES DATA MODEL DESIGN

This data model was developed for a CPR system named GERI at the BVAMC. The clinic has a defined workflow for placing volunteer research candidates into one of four research demonstration projects involving smoking cessation, exercise programs, nutrition, and stroke rehabilitation. The volunteer research candidates (hereafter referred to as patients) consist of veterans who have one or more cardiac risk factors and meet the criteria for one or more research protocols.

We chose a database development system from Gupta Technologies Inc. (the SQLWindows development system for the MSWindows client and the SQLBase Netware database server). This product is an RDBMS with client/server network connections.

The core of the GERI CPR is the care planning module, which is the clinical interface to the system. Figure 1 shows this interface. It allows clinicians and researchers from various disciplines to see all patient information. Dynamic views, order entry, and reporting are supported. The remainder of this section describes relevant portions of this interface in order to show the level of complexity in the transferred patient data.

Fig. 1. The Main GERI Window.



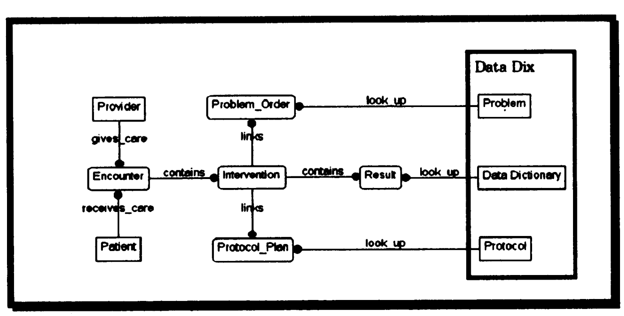
The CPR is encounter-based in that all displayed information is specific to the selected encounter. The separate table window *Interventions* at the bottom of the GERI window in Figure 1 lists all patient events for the selected encounter. Intervention is a broad term that is here used to describe any direct or indirect care or other documented event for a patient in an encounter. It includes the carrying out of typical orders, surveys, examinations, procedures, etc. For example, the patient had a functional status assessment on 3/31/94. The user can control the content of the bottom window in Figure 1. For example, the user can view current problems, or pending orders in that window instead of interventions.

An important feature of this care plan is the fact that all orders (or non-orders) are linked to problems (or a protocol) in the record [4]. This allows more detailed information for quality assurance, billing, and clinical research. "Accurate measures of resource consumption also constitute the bridge between the use of an AES [ambulatory encounter system] for traditional quality or payment purposes (or both) and the current push toward continuous quality improvement (CQI)." [5]

This interface is based on the simplified data model in Figure 2 (the actual data model has more than 20 entities and the data dictionary is a more complex multi-entity object). The data model that produced this interface revolves around the concept of a patient *encounter*. This is an encounter between a patient and a provider of some type.

The complex information contained in this data model is captured with the graphical user interface (described above) that allows linking of data elements and simultaneous data element browsing. This data complexity is captured economically in the data model. The argument for Proposition 1 is that the information density of this kind of patient information is possible to transfer with a standard model.

Fig. 2. The Simplified Data Model



This case study shows the complex interrelated data elements required for AESs. Without a model, all assumptions about the relationships in the data would have to be contained in the transfer document and there may be an impedance mismatch with the target database. This section has shown that certain aspects of complex patient information such as the time-oriented, protocol-driven, and linked data of this case can be represented economically using the data model. The issue of economy of representation leads to Proposition 2 and is discussed below.

PROJECT METHODOLOGY

The methodology used in this project to transfer patient records between sites is dependent upon semi-standard SQL-based RDBMSs and a common portion of the data model described above. Simple programs unload patient data from the relational tables into text script files that can be run at the receiving site to update their database. This methodology requires a minimum of program development and user expertise, but relatively extensive standardization. It is our position that this standardization is worth the effort in order to reap the demonstrated benefits of information density, efficiency, and ease of communications. Whether such standardization is politically or logistically possible in the current healthcare information systems context is not in the scope of this paper, although it is now a key theme of healthcare informatics [6]. The transfer methodology has the following steps.

Sending Database Task Steps:

1. Extract patient information from each table of the sending database.
2. Create a text script file to update the receiving database.
3. Compress and encrypt the script file with a password.
4. Log the transaction.

Receiving Database Task Steps:

5. Uncompress and unencrypt the script file.
6. Run the script against the receiving database.
7. Log the transaction.

The database at each site must have a portion of their data model in common. Each site may have additional (non-shared) tables and additional (non-shared) attributes in the shared tables, but they must have a core of tables and attributes exactly in common. Because of this flexibility, the only real tool restriction with this shared model is the requirement of a RDBMS from any major vendor. The extraction program (Step 1) is very simple. Every table in the model is queried for any rows linked to a particular patient ID (and possibly in a particular date range). This requires a simple SQL query for each table in the model. All of this data is written to a single text file that becomes a SQL script for updating the receiving database. The script file (Step 2) is a sequence of SQL queries that are automatically run one after another. For example, the following 'insert' statement would be one of many in the script (one for each table in the shared model).

```
INSERT INTO SYSADM.PROBLEM
VALUES(:1,:2,:3,:4,:5,:6,:7,:8,:9,:10,:11)
\
5,"DIAM","",44,10002,1993-06-10,,"O",,,$long,
6,"ATH","",40,10002,1993-06-10,,"O",,,$long,
7,"ATH","",81,10022,1993-06-23,,"O",,,$long,
8,"STOC","",79,10002,1993-06-23,,"O",,,$long,
10,"STOC","",81,10022,1993-06-23,,"O",,,$long,
/
```

Note that this implementation of the 'insert' command supports bind variables. Each bind variable (such as ':1') corresponds by position with the comma separated block of text delimited with the forward and backward slashes. Every line of this text block is inserted as a record into the receiving table with this one SQL statement. This is a very non-standard feature across RDBMSs and would require the most development and coordination efforts. Imports, exports, and the SQL Data Definition Language (DDL) are the most non-standard elements of RDBMSs and SQL. Both sites in this demonstration project used Gupta's SQLBase database server and therefore did not deal with this standardization problem. This required development for multivendor interoperability is minor compared to methods that do not share a data model, common SQL Data Control Language (DCL), and relational database implementation.

The script resulting from these two steps is a (human-readable) text file that is sent by the sending site. The receiving site simply runs this script file against the database that is updated with the transferred patient(s') data. All linked information and referential integrity are preserved because the data model is replicated at each site. Compression and encryption of the script file (Step 3) can be performed with any applicable utility. The only requirement is that both the sending and receiving sites have the same program. Password access to the encrypted file secures the patient data to the limits of the encryption scheme and the security procedures of the provider sites. We used the shareware utilities PKZIP and PKUNZIP for this step. This utility may not be very secure and therefore not adequate for real operations, but it precisely demonstrates the concept. The sender's log file (Step 4) records any standard information that each site requires such as the identification of the sender and any notes relating to the transfer. This log may update a table on the senders database and a copy of this log is sent with the data to the receiver.

The receiving site must uncompress and unencrypt the script file (Step 5) and run the script against their database (Step 6). In the case of Gupta's RDBMS, Step 6 is accomplished simply by using a 'Run Script' option that sequentially executes any SQL statements in a script file. This utility is offered by many vendors and would be a relatively simple utility to add to any RDBMS. The sender's log file can automatically update the receiver's log table (Step 7), documenting the transaction.

A crucial factor in sharing patient data is a common data dictionary for healthcare terminology. This is a difficult problem and this methodology allows a relatively simple way to share a standard data dictionary between sites. Once a truly standard terminology is available, it could be substituted. Script files can be created with the same methodology as above to transfer the data dictionary information between sites. In this demonstration project, both sites use the same data dictionary.

Relational database schemas to create patient databases are also easily transferred as scripts in this same way. There are many data modeling CASE tools on the market that allow a developer to graphically define a data model and then automatically generate a SQL script that will create the tables and indexes for that model in a target RDBMS. Because of the non-standard nature of current DDLs, these tools typically generate a custom script for the RDBMSs of most major vendors.

DEMONSTRATION PROJECT RESULTS

This demonstration project was performed to test support for the last two of the three initial propositions. It is not a controlled experiment, but a proof of concept exercise. In order to test the methodology for efficiency, reliability, and ease of communication, we developed an application that implements the seven steps in the patient record transfer methodology. This application works within the GERI patient record application as a menu item. It performs the steps in the methodology in a way that requires minimal training of personnel already using GERI. The user initiates a patient record transfer in the context of a displayed patient record by choosing a menu item 'Transfer' under the 'File' menu and sees a dialog box where all transfer actions are performed.

The user specifies a script filename and emails a log note for the transfer. For this demonstration project, there was only one destination site and therefore that did not need to be specified. The transfer application creates and compresses the script file with a password, and the initiator sends this file (by FTP) to the receiving site. At the receiving site, the receiver uses a batch file to uncompresses the script and run it against the database. An email message from the sending site notifies responsible users of the transfer and sends the log information. An email message from the receiving site acknowledges the transfer.

The demonstration project tested this application of the transfer methodology with a one week trial. Real patient data (stripped of identifying information such as name and SSN) was sent from the BVAMC (sending site) to LHI (receiving site) over the Internet. A person (at each site) was responsible for sending the information from Baltimore and updating the database at LHI. These tasks required no knowledge of the database or the methodology other than that necessary to perform the duties. The person at the sending site received an email message asking to send a record for a particular patient. This person then enters GERI, opens that patient's record and uses the transfer application. The person at the receiving site opens the transfer application (outside of GERI in a SQL tool) and updates the database. This trial transferred only entire patient records (all records corresponding to one patient) and did not support specific date ranges.

The trial had 3 phases. The first phase transferred a script from the BVAMC to LHI that created the database. This step simulates setting up a database at a repository site, a consultant site, or remote site for a clinic. This was accomplished in one step with a script that created the tables, the indexes, and the stored commands. This task requires no database experience other than the knowledge of how to run the script. The second phase transferred a script that contained the data dictionary tables. This step simulates providing updates of the data dictionary for remote sites to keep all sites in synch. The third phase transferred 10 patient records between the BVAMC and LHI. This process was conducted over a week-long period. Email was used for notification and logging of each transaction. The first two transfers required phone support to complete the training. All of the remaining eight transfers were successful and required no additional support. This trial demonstrates that this methodology is logistically simple to implement and does not require large support resources.

DISCUSSION AND EVALUATION

Each of the propositions discussed in this paper assume that the patient record transfer methodology is driven by computer-based clinical automation techniques that integrate well into current database, communications, and CASE tool technologies. Techniques that do not integrate well with current technologies will not be adopted due to cost and scarcity of expertise. As database architectures and their tools change, the specific methodology, such as that given here, would also change, but the three propositions would stand. The following conclusions about the three initial propositions are supported by the demonstration project.

The demonstration project shows that complex patient data (data with relationships) can be transferred easily between environments with a common subset of a standard data model. This standard data model is a higher level standard than HL7 in the sense that HL7 is a standard for exchanging transaction-oriented messages between (possibly) heterogeneous systems, while the standard data model approach to patient record transfer is a systems-oriented standard that assumes more than a common terminology for transactions. It assumes standard relationships. The Joint Working Group for a Common Data Model has begun working on just these kinds of standards [7]. Their work can form a basis for this type of inter-site communications.

This study has assumed that this data model is implemented in an RDBMS but this is not a requirement. Other more general (than SQL) representation systems can and will be developed. For example, the SQL scripts can be wrapped in a language/protocol such as Knowledge Query and Manipulation Language (KQML) [8] to supply the transport and performatives for use on an internetwork using intelligent agents. The content language may change from SQL to KIF (Knowledge Interchange Format) for greater flexibility and generality, but the basic concepts of this study would remain.

The methodology described here is efficient because it offers integration with the user interface, uses existing standards, co-opts the work of vendors, and offers ease of development. These methods requires network resources that are available over the Internet or other private networks such as Community Health Information Networks (CHINs). Currently, stable technology and network infrastructure exist to support extensive patient care repositories for research, DSS, and education. They also have relevance to clinical care. For example, the methods described here are easily modified to produce executable single patient versions of the CPR that could be sent over the network as a self-contained CPRs to providers doing consults. The methodology is not, however, currently reliable because of known problems with TCP/IP transport and lack of a principled procedure for tracking the log files and other inter-site communications. The reliability of the Internet transport is now being aggressively addressed in research and commercial areas. The log file procedure to ensure database integrity is an area for future work.

The barriers to the benefits of these methods are also due to organizational issues and communications problems. It is a complex management challenge to deploy information systems in an organization. It is even more difficult for inter-organization standards to be agreed to and implemented. This paper has developed and tested a method for CPR transfer that is designed to soften these very difficult problems. We are planning a more extensive and realistic test of this methodology between the VAMCs at both Baltimore and Seattle.

Reference

[1]. McDonald, C.J. & Barnett, G.O. in Medical Informatics: Computer Applications in Health Care, eds. Shortliffe, E., et al., Addison-Wesley:Reading MA, 1990, pp. 181-218.

[2]. Henrion, M., Silva J., Cost Savings from Information Technology in the US Health Care Reform: Insights from Modeling, *Healthcare Information Management*, 8(1), 1994 pp. 23-28.

[3]. Bruce, T.A. Designing Quality Databases with IDEF1X Information Models. Dorset House:New York, NY, 1992.

[4]. Beeler, G., Gibbons, P., & Chute, C. Development of a Clinical Data Architecture. In M. Frisse (Ed.), 16th SCAMC, McGraw-Hill:Baltimore, 1992, pp. 244-248.

[5]. Goldfield, N. Ambulatory encounter systems: Implications for payment and quality. *J. Ambulatory Care Manage,* 16(2), 1993, pp. 33-49.

[6]. Hammond, W., McDonald, C., Beeler, G., Carlson, D., Barnett, L., Bidgood, D., & McDougall, M. Computer Standards: Their Future within Health Care Reform. In Proceedings of the 1994 Annual HIMSS Conference, Phoenix AZ, 1994.

[7]. Beeler, G. HISPP/MSDS Joint Working Group for a Common Data Model Report of Meeting Sept. 20-21, 1993.

[8]. Finin, T., et al. Specification of the KQML Agent-Communication Language. Technical Report EIT TR 92-04, Enterprise Integration Technologies, Palo Alto, CA, 1992.